MODELOS DE APARÊNCIA ATIVA UTILIZANDO NÍVEIS DE CINZA, GABOR JETS E LBP PARA VERIFICAÇÃO DO USO DO CAPACETE DE MOTOCICLISTAS

Felipe José Aguiar Maia*, José Everardo Bessa Maia†, Thelmo Pontes de Araujo‡

* *Universidade Estadual do Ceará, Av. Dr. Silas Munguba, 1700, Fortaleza, 60714-903, Brazil*

Emails: felipe.ja.maia@gmail.com, jose.maia@uece.br, thelmo.dearaujo@gmail.com

Abstract— This work looks into the problem of detecting the presence of motorcyclist safety helmet on single images of city roads. The degree of adjustment of a new image to the previously trained Active Appearance Model (AAM) acts as the decision criterion for detection. This adjustment is measured by differences between AAM shape and appearance parameter vectors, which become the feature vectors for four different classiffers. The AAMs were generated using three possible texture information: shades of gray, Gabor features and LBP features. Results show that this approach is feasible and promising.

Keywords— Artificial Intelligence, Image Processing, Pattern Recognition, Image Recognition, Learning Models.

Resumo— Este trabalho investiga o problema de detectar a presença do capacete de segurança de motociclistas em imagens únicas de vias urbanas. O grau de ajustamento de uma nova imagem com os Modelos de Aparência Ativa (AAM) anteriormente treinados atua como o critério de decisão para a detecção. Este ajuste é medido pela diferença entre os vetores que contêm os parâmetros de forma e aparência do AAM, que se tornam os vetores de características para quatro classificadores diferentes. Os AAMs foram construídos utilizando três possíveis características da textura: os tons de cinza, as características de Gabor e os padrões binários locais (LBP). Os resultados mostram que esta abordagem é viável e promissora.

Palavras-chave— Inteligência Artificial, Processamento de Imagens, Reconhecimento de Padrões, Reconhecimento de Imagem, Modelos de Aprendizagem.

1 Introdução

Algoritmos para resolver o problema da detecção de múltiplas instâncias de um objeto em uma determinada imagem e verificar suas propriedades ainda são um desafio, especialmente em cenários muito genéricos. Variações de iluminação, silhueta e fundo são algumas das dificuldades frequentemente encontradas. Este breve artigo tem como objetivo verificar a presença do capacete de segurança de motociclistas em imagens estáticas de vias urbanas.

Imagens de vias urbanas obtidas por sistemas de monitoramento de trânsito geralmente têm má qualidade (devido às variações de iluminação, chuva etc.). Por essa razão, foram escolhidas características obtidas através dos Modelos de Aparência Ativa (AAM) para a classificação, uma vez que esses modelos funcionam bem com formas pouco definidas (Cootes et al., 1998).

O problema de localização e rastreamento de motociclistas em vídeos de vias públicas é abordado por muitos autores. Em (Silva et al., 2013b) são utilizadas características de Haar, Histogramas de Gradiente e Padrões Binários Locais (LBP) como características a serem classificadas por classificadores como a rede neural de Função de Base Radial (RBF), a rede neural Perceptron de Múltiplas Camadas (MLP), e *Support Vector Machine* (SVM). O fundo (*background*) das imagens é suprimido para reduzir o processamento. Esse trabalho foi estendido em (Silva et al., 2013a) para determinar a presença de capacetes de segurança nas regiões mais prováveis de se encontrar a cabeça do motociclista.

Em (Ku et al., 2008), uma pesquisa por contornos similares à círculos é feita a fim de detectar capacetes de segurança em motociclistas localizados em imagens de vias. Esse trabalho considera também a presença de oclusões.

O trabalho (Chiverton, 2012) localiza veículos em movimento após a remoção do *background* da imagem. Então, motos e carros são classificados a partir de suas proporções. Os histogramas das regiões mais prováveis para se encontrar a cabeça do motociclista são usados para determinar a presença ou não do capacete de segurança.

A maioria dos trabalhos nesta área ((Ku et al., 2008); (Silva et al., 2013a); (Chiverton, 2012)) utilizam sequências (*streams*) de vídeo como entrada de dados em vez de imagens únicas de cada veículo. Esta pesquisa busca resolver o problema através de uma única imagem do motociclista, uma vez que é interessante a utilização da verificação do capacete em conjunto com outros processos de detecção de infrações, como por exemplo violação de semáforo vermelho ou veículo acima da velocidade permitida.

Neste trabalho, os vetores de parâmetros de forma e aparência obtidos a partir de um Modelo de Aparência Ativa são utilizados como características para verificar a presença de capacete de segurança em motociclistas usando um classificador supervisionado. Durante a geração dos Modelos de Aparência Ativa, foram utilizadas três diferentes características para representar a textura: os tons de cinza, as características de Gabor e as características LBP.

Note que o vetor de níveis de cinza é uma representação analógica no espaço de imagens, o LPB é um vetor de atributos binários de textura local e jatos de Gabor são atributos localizados no domínio da frequência (wavelets).

"Presença de capacete de segurança" é escolhida como a classe objetivo, uma vez que a forma e a textura artificial dos capacetes são em geral mais distinguíveis do que as cabeças humanas sem um capacete.

Este artigo possui a seguinte estrutura: a Seção 2 mostra como vetores AAM são obtidos. A Seção 3 descreve a metodologia. Resultados e discussão estão na Seção 4. A Seção 5 conclui o trabalho.

2 Modelos de Aparência Ativa

Os Modelos de Aparência Ativa (AAM) foram propostos por (Cootes et al., 1998) como uma extensão para os Modelos de Forma Ativa (ASM) (Cootes et al., 1995) e são capazes de modelar tanto forma quanto aparência. Os AAMs apresentam maior robustez em relação aos ASMs por utilizarem um conjunto mais completo de informações da imagem (Edwards et al., 1998). A forma é definida como um conjunto de pontos de controle ordenados e rotulados com a intenção de capturar a forma espacial do objeto (e também de suas partes). Já a aparência está relacionada com textura do objeto, que normalmente é representada pelos níveis de intensidade dos tons de cinza (ou cores) dos pixels na imagem do objeto.

A primeira tarefa na construção de um AAM ou ASM é a criação de um Modelo de Distribuição de Pontos (PDM). Isso é feito otimizando o alinhamento (por translação, rotação e escala) dos pontos de controle do conjunto de treinamento com a sua forma média.

Após o alinhamento dos pontos do conjunto de treinamento da base de dados, esses pontos formam nuvens onde pode-se notar a variabilidade de cada um desses pontos de interesse, como pode ser visto na Figura 1.

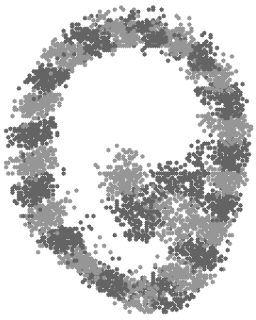

Figura 1: Pontos do conjunto de treinamento após serem alinhados.

A Análise de Componentes Principais (PCA) (Jolliffe, 2002) é então aplicada nos pontos do conjunto de dados alinhados para obter um modelo capaz de descrever exemplos de formas do objeto a partir de um vetor de parâmetros b_s:

$$x = \bar{x} + P_s b_s \,, \tag{1}$$

sendo $\bar{x}$ a forma média, P_s a matriz que contém os principais autovetores (ortonormais) da matriz de covariância do conjunto de dados, e b_s o vetor que descreve a forma x.

Uma vez obtida a forma média, um algoritmo de triangulação de Delaunay (como o de divisão e conquista mostrado em (Guibas and Stolfi, 1985)) é aplicado aos pontos dessa forma média (Figura 2, meio). Essa mesma triangulação é reproduzida em cada imagem rotulada do conjunto de treinamento (Figura 2, esquerda). Finalmente, os pixels em cada região triangular das imagens de treinamento são mapeados nos triângulos correspondentes na forma média, criando uma imagem distorcida que casa com a forma média já obtida (Figura 2, direita).

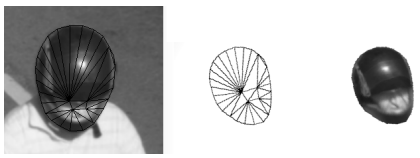

Figura 2: Exemplo do imagem do conjunto de treinamento com a sua textura distorcida para a forma média.

Os pixels em cada imagem mapeada são normalizados para reduzir o efeito da variação da luz e em seguida são transformados em vetores. A PCA é aplicada nesses vetores para obter um modelo linear da textura:

$$g = \bar{g} + P_g b_g \,, \tag{2}$$

sendo $\bar{g}$ a média dos vetores de níveis de cinza do conjunto de treinamento normalizados, P_g a matriz que contém os principais autovetores (ortonormais) da matriz de covariância dos dados de textura, e b_g o vetor que contém os parâmetros para geração da textura g.

Os vetores b_s e b_g descrevem, respectivamente, a forma e a textura de cada imagem do objeto. Para dirimir possíveis correlações entre esses vetores, a PCA é aplicada uma terceira vez:

$$b = \begin{bmatrix} W_s b_s \\ b_g \end{bmatrix} = \begin{bmatrix} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{bmatrix} , \tag{3}$$

sendo que a matriz diagonal W_s compensa a diferença entre as unidades de forma e de textura, obtendo-se:

$$b = Qc, \text{ onde } Q = \begin{bmatrix} Q_s \\ Q_g \end{bmatrix} , \tag{4}$$

sendo as colunas ortonormais de Q formadas pelos principais autovetores e c o vetor de aparência que descreve tanto a forma quanto a textura.

Como o modelo é linear, forma e textura podem ser descritas separadamente por c utilizando:

$$x = \bar{x} + P_s W_s Q_s c \quad \text{e} \quad g = \bar{g} + P_g Q_g c\,. \tag{5}$$

Dados uma nova imagem e o modelo de aparência previamente descrito, os parâmetros do modelo podem ser ajustados de maneira a alinhar o melhor possível a imagem sintetizada pelo modelo e a nova imagem. Isso pode ser tratado como um problema de otimização em que o quadrado da norma da diferença entre a nova imagem e a imagem sintetizada deve ser minimizado:

$$min \parallel \delta I \parallel^2, \quad \text{com } \delta I = I_i - I_m\,, \tag{6}$$

sendo I_i o vetor que contém a representação da textura da nova imagem e I_m contém a representação da textura da imagem sintetizada pelo modelo de aparências.

Idealmente, o valor mínimo na Equação (6) é obtido variando-se o vetor de parâmetros de aparência c (que gera I_m) na Equação (5), porém esse procedimento é muito custoso. Utiliza-se, então, regressão para encontrar uma relação linear entre as variações no vetor de parâmetros δc e δI:

$$\delta c = A \delta I\,. \tag{7}$$

Imagens sintéticas podem ser geradas (usando as equações (5)) perturbando-se o vetor de parâmetros c de imagens no conjunto de treinamento. Pequenas perturbações de escala e pose também podem ser adicionadas. É possível computar a matriz A seguindo os seguintes passos:

1. A partir de uma imagem de treinamento original, c_0 pode ser determinado e então perturbado por um dado δc para se obter $c = c_0 + \delta c$;
2. Os vetores de forma e textura x e g_m podem ser computados para o vetor perturbado c;
3. A imagem original é deformada para casar com o vetor de forma x e um novo vetor de textura g_s é computado;
4. A diferença na aparência é calculada por $\delta g = g_s - g_m$;
5. A regressão $\delta c = A \delta g$ fornece a matriz A.

Após descobrir a matriz de conhecimento *a priori* A, o procedimento de ajuste do modelo pode ser executado em uma nova imagem fora do conjunto de treinamento. Dada uma nova imagem, uma região de interesse (ROI) é localizada e os parâmetros de pose, escala e rotação do PDM são ajustados de acordo. Em seguida, o Algoritmo 1 é aplicado.

Algoritmo 1: Ajuste do modelo em uma nova imagem.

Entrada: $\bar{x}$; $\bar{g}$; P_s; W_s; Q_s; P_g; Q_g; A; vetor contendo a textura da nova imagem g_s.
Saída: Parâmetros do modelo que melhor descrevem a imagem: c; vetores de aparência: g e s.
início
  Inicialize c_0 , $g_0 = \bar{g}$, ϵ, $\Delta\epsilon$,$\Delta\epsilon_{max}$;
  enquanto $\Delta\epsilon > \Delta\epsilon_{max}$ **faça**
    $\delta g = g_s$ - g_0; $\epsilon_0 = \parallel \delta g \parallel^2$;
    $\delta c = A\delta g$;
    $\eta = 2$;
    enquanto $\epsilon > \epsilon_0$ *ou* $\eta > 0.001$ **faça**
      $\eta = \eta/2$;
      $c = c_0 - \eta\delta c$;
      $g = \bar{g} - P_g Q_g c$;
      $\delta g = g - g_s$;
      $\eta = \parallel \delta g \parallel^2$;
    fim
    Atualize: $g_0 = g$; $c_0 = c$; $\Delta\epsilon = |\epsilon - \epsilon_0|$; g_s;
  fim
  Compute: $x = \bar{x} + P_s W_s Q_s c$;
  $g = \bar{g} + P_g Q_g c$.
fim

3 Metodologia

Neste trabalho, a metodologia é composta de três etapas: detecção da ROI, extração de características e classificação (Figura 3).

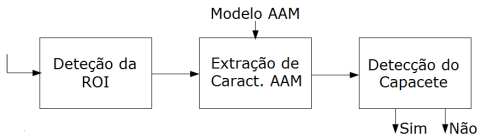

Figura 3: Fluxo de execução.

Diferentemente de trabalhos que utilizam *streams* de vídeo ((Ku et al., 2008); (Silva et al., 2013b), (Chiverton, 2012)), este trabalho utiliza apenas imagens estáticas para determinar a presença do capacete de segurança. Para detectar a presença do motociclista (com e sem capacete) nos experimentos, o algoritmo Viola-Jones – baseado em características de Haar e AdaBoost – foi escolhido por conta de seu poder de generalização e robustez (Viola and Jones, 2001). O detector foi treinado até que uma taxa razoável de falso negativo foi alcançada (15,76%). A Figura 4 mostra detecções positivas da ROI em ambos os casos: sem (a) e com (b) o capacete de segurança.

A qualidade da detecção da ROI é um fator importante para o sucesso do algoritmo de convergência do AAM. Em (Cootes et al., 1998), é mos-

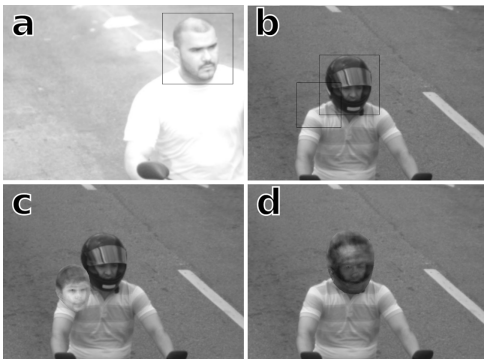

Figura 4: Detecção de ROI e de faces.

trado que os deslocamentos computados durante o algoritmo AAM devem ser pequenos o bastante para garantir a linearidade da relação entre o erro e o ajuste do modelo, isso é necessário para o correto uso da regressão linear. No caso em que uma região de interesse não é encontrada, a execução é abortada.

Um vez que a ROI é encontrada, o algoritmo AAM é executado para extrair as características da imagem. Além dos níveis de intensidade de cinza que são normalmente utilizados na construção dos modelos, foram realizados testes gerando modelos utilizando características de Gabor e características LBP para representar a textura na imagem. Nesse caso, em vez dos tons de cinza, o vetor contendo as caraterística de textura de cada pixel é concatenado ao vetor g, que representa a textura.

É possível observar um exemplo do resultado da execução do algoritmo de convergência do AAM na Figura 5. Nesse caso, temos a imagem original e uma aproximação gerada pelo modelo usando um conjunto pequeno de características ($n = 206$).

Figura 5: Convergência do AAM usando tons de cinza: caso com capacete.

Para descartar os falso positivos, em casos de detecção dupla, é escolhido o vetor de parâmetros mais próximo à média do modelo correspondente, assim eliminando detecções indesejadas (Figura 4 (b)(c)(d)).

Na fase de classificação, quatro classificadores foram comparados: Support Vector Machine (SVM) com núcleo Polinomial, SVM com núcleo Sigmoide, a rede neural Perceptron de Múltiplas Camadas (MLP), e árvores de decisão criadas através do algoritmo C4.5 (Quinlan, 1993).

Na seção a seguir, os resultados mostram a performance de cada classificador considerando o vetor de características extraído pelo algoritmo AAM. A saída de cada classificador é sempre a classe "com capacete" ou a classe "sem capacete".

4 Resultados

4.1 Experimentos

Os testes foram implementados utilizando a linguagem C++ e a biblioteca de visão computacional OpenCV (OpenCV, 2014). A base de dados utilizada nos experimentos foi composta de 1060 imagens, 530 de cada classe. Para o treinamento do detector de Viola-Jones foram utilizadas 800 imagens (400 de cada classe) do conjunto total. As demais 260 imagens foram usada para treinamento e teste da fase de extração de características e classificação.

Do total de exemplos, 551 foram colhidas em vias urbanas, sendo que 530 eram imagens de motociclistas com capacete e apenas 21 eram de motociclistas sem capacete. Essa desproporção ocorre porque é raro encontrar motociclistas sem capacete trafegando dentro da zona urbana de uma cidade grande. Para solucionar essa disparidade foram utilizadas imagens da base de imagens Yaleface database (Georghiades et al., 2001). Essa base foi escolhida por conter imagens de cabeças em diversas poses e ângulos, inclusive em ângulos semelhante aos de imagens que seriam capturadas por uma câmera de trânsito. Assim, foram selecionadas aleatoriamente 509 imagens nas poses 4, 6 e 7, totalizando as 1060 imagens sendo 530 de cada classe. Esse balanceamento teve como objetivo facilitar a análise de sensibilidade dos classificadores.

Foram modificados apenas os seguintes parâmetros no treinamento do algoritmo Viola-Jones do OpenCV: taxa de falso positivo máxima de 0,01; 8 estágios de treinamento; profundidade máxima de 4 camadas; contagem máxima de 5 classificadores fracos; tamanho da janela de 24×24. O classificador em cascata utilizou as características de Haar. Foi feito somente um treinamento preliminar do Viola-Jones, já que um definitivo depende de uma seleção criteriosa dos dados e foi deixado para um trabalho em preparação.

Um esquema de validação cruzada de 10 *folds* foi utilizado para treinar e testar os classificadores usando as 260 imagens. Todas as imagens foram rotuladas manualmente com 30 pontos de controle para a construção dos modelos de distribuição de pontos. As PCAs executadas na construção do modelo de aparências mantiveram 99% da variabilidade, por esse motivo número de características extraídas pelo AAM varia de acordo com o conjunto de dados.

Para a construção do vetor de textura usando características de Gabor, foram usadas 4 variações

de ângulo (0π, $\frac{1}{4}\pi$, $\frac{1}{2}\pi$, $\frac{3}{4}\pi$) e 3 variações comprimento de onda (1, 4, 8), e um total de 12 núcleos, todos de tamanho 3×3.

A matriz de regressão que aproxima o modelo foi construída realizando 50 perturbações aleatórias em cada exemplo do conjunto de treinamento, cada perturbação com ± 2 vezes o desvio padrão correspondente.

Para os classificadores SVM, foi utilizado o fator de penalidade $C = 0{,}1$. O núcleo sigmoide utilizado foi $k(x, y) = \tanh(\gamma x^T y + c)$, onde $\gamma = 10^{-5}$, $c = 274$. O núcleo polinomial foi $k(x, y) = (\gamma x^T y + c)^d$, onde $\gamma = 10^{-5}$, $c = 274$ e $d = 0{,}07$. Quanto à MLP, foi utilizada uma topologia contendo uma única camada escondida com o dobro do número de parâmetros (variável) como nerônios, taxa de aprendizagem de 0,1 e *momentum* de 0,1.

Para execução os classificadores SVM e a rede neural MLP foi utilizada a implementação da biblioteca OpenCV. Para a construção das árvores de decisão, foi utilizada a implementação J48 presente na biblioteca WEKA (Hall et al., 2009).

A saída positiva do classificador ocorre quando o motociclista está utilizando o capacete, enquanto que a negativa ocorre quando ele está sem o capacete de segurança. Dessa forma, durante a análise das tabelas de confusão dos classificadores, a taxa de verdadeiro positivo (TVP) indica a proporção de exemplos com capacete classificados corretamente. A taxa de falso negativo (TFN) indica a proporção de exemplos com capacete classificados incorretamente. A taxa de verdadeiro negativo (TVN) indica a proporção de exemplos sem capacete classificados corretamente. A taxa de falso positivo (TFP) indica a proporção de exemplos sem capacete classificados incorretamente.

4.2 Resultados e Discussão

Após aplicar o algoritmo de Viola-Jones, foram selecionadas as detecções verdadeiro positivo que seguiram para a fase de classificação, cujos resultados são mostrados na Tabela 1.

A média e variância da acurácia usaram como base a taxa de acerto de cada execução da validação cruzada. Para as demais taxas, as matrizes de confusão foram somadas e usou-se:
$TVP = \frac{\sum \text{Verd. Pos.}}{\sum \text{Classif. Pos.}}$; $TFN = \frac{\sum \text{Falsos Neg.}}{\sum \text{Classif. Pos.}}$; $TVN = \frac{\sum \text{Verdad. Neg.}}{\sum \text{Classif. Neg.}}$; $TFP = \frac{\sum \text{Falsos Pos.}}{\sum \text{Classif. Neg.}}$.

Como pode ser visto na Tabela 1, os resultados da árvore de decisão J48 manteve a taxa de acerto entre 74% e 76% dependendo da característica utilizada para representar a textura. A rede neural MLP apresentando melhor desempenho com as características de Gabor (81,65%) e pior desempenho com as características LBP (73,5%). Sendo assim, é interessante observar a consistência do desempenho desses classificadores independente das características escolhidas para representação da textura.

Tabela 1: Resultados dos classificadores.

Classif. (caract. textura)	Acur. $\mu(\sigma)^2$	TVP	TFN	TVN	TFP	AUC
Níveis de Cinza						
SVM Poli.	0,8186 (0,0037)	0,9224	0,0776	0,6990	0,3009	0,8970
MLP	0,7846 (0,0109)	0,8534	0,1466	0,7087	0,2912	0,8971
J48	0,7500 (0,0082)	0,8534	0,1466	0,6311	0,3689	0,7422
SVM Sigm.	0,5993 (0,0202)	0,3793	0,6207	0,8349	0,1650	0,6358
Características de Gabor						
SVM Poli.	0,5918 (0,0029)	0,3534	0,6465	0,8641	0,1359	0,5950
MLP	0,8165 (0,0055)	0,8965	0,1034	0,7281	0,2718	0,9106
J48	0,7556 (0,0051)	0,8534	0,1465	0,6407	0,3592	0,7471
SVM Sigm.	**0,8374 (0,0112)**	0,9483	0,0517	0,7087	0,2913	0,9090
Características LBP						
SVM Poli.	0,8179 (0,01254)	0,8835	0,1165	0,7340	0,2660	0,8715
MLP	0,7350 (0,0077)	0,8534	0,1465	0,6019	0,3981	0,7289
J48	0,7465 (0,0093)	0,8362	0,1637	0,6505	0,3495	0,7433
SVM Sigm.	0,5133 (0,0071)	0,1465	0,8534	0,9223	0,0777	0,6945

No caso dos classificadores SVM com núcleo polinomial e sigmoide, do desempenho variou mais dependendo das características escolhidas. O classificador SVM polinomial apresentou melhores taxas de acerto com as características LBP (81,79%) e os tons de cinza (81,86%), porém apresentou desempenho inferior quando utilizando as características de Gabor (59,18%). O classificador SVM sigmoide apresentou diferente resultado, com a melhor taxa média de acerto com as características de Gabor (83,74%) e desempenho inferior com as características LBP (51,33%) e os tons de cinza (59,93%).

O classificador SVM sigmoide utilizando características LBP apresentou a pior precisão média, apesar de alcançar a melhor (mais baixa) taxa de falsos positivos (TFP = 7,77%). Isso poderia ajudar as autoridades de trânsito a punir motociclistas que deixam de usar o capacete de segurança (uma coisa boa), porém isso aumenta o trabalho de verificação humana do sistema, pois apresenta uma taxa de falso negativo (TFN) de 85,3%, o que geralmente não é aceitável.

O SVM com núcleo sigmoide com características de Gabor apresentou a melhor taxa de acerto média entre todos os classificadores e todas as caraterísticas de textura, com 83,74%. Ele também obteve a melhor (maior) taxa de verdadeiro positivo (94,83%). Em um sistema automatizado para auxiliar na fiscalização do trânsito, um baixo TFN

(5,17%) é desejável, pois diminui o trabalho de verificação humana (uma vez que punições podem ser aplicadas). Porém, normalmente isso reflete na TVN (70,87%), gerando alguma perda na detecção de motociclistas infratores.

O SVM polinomial utilizando características LBP apresentou uma TVN ligeiramente melhor de (73,40%), porém com uma TVP inferior de 88,35%. Sua taxa de acerto foi pior que o SVM sigmoide com características de Gabor, sendo de 81,79%.

A relevância das métricas na Tabela 1 depende do objetivo da aplicação: se o objetivo é encontrar motociclistas sem o capacetes de segurança, TVN e TFP são as métricas relevantes. Entre os classificadores com taxa de acerto superior a 70%, o classificador SVM polinomial com características LBP se destacou, apresentando melhor TVN = 73,4% e TFP = 26,6%. Além disso, os classificadores SVM polinomial com características de Gabor e SVM sigmoide com tons de cinza apresentam uma taxa de acerto média próxima a 60%, com uma TVN bem superior, entre 83% e 87%.

Quando o objetivo é reduzir o esforço de verificação humano, é importante observar as taxas de verdadeiro positivo e falso negativo. Nesse caso o classificador SVM com núcleo sigmoide de características de Gabor alcançou o melhor desempenho, com TVP = 94,83% e TFP = 0,05%.

Em termos de precisão média, o desempenho dos classificadores SVM polinomial com tons de cinza e características LBP e o classificador SVM sigmoide com características de Gabor é praticamente o mesmo, com uma pequena vantagem para o SVM sigmoide Gabor (precisão média = 83,74%).

As curvas ROC, nas Figuras 6, 7 e 8, nos permite uma melhor avaliação do *trade-off* entre os dois tipos de erro em todos os classificadores. Elas também permitem que, dada uma TFP, seja possível estimar o TVP de cada classificador. Por exemplo, se um TFP = 20% é admissível para a aplicação, as curvas ROC mostram que o classificador SVM polinomial com tons de cinza e e SVM sigmoide Gabor alcança uma TVP próxima de 90%.

Os valores da área sob a curva ROC (AUC) (última coluna na Tabela 1) mostram resultados muito semelhantes para três classificadores destacados (SVM polinomial com tons de cinza, SVM polinomial LBP, e SVM sigmoide Gabor), de modo AUC não deve ser utilizado como o único critério de desempenho. Porém, é interessante notar que o classificador MLP com características de Gabor obteve a melhor AUC entre todas as combinações testadas, esse valor se deve ao seu desempenho superior a todos os outros quando a TVP é menor que 60%.

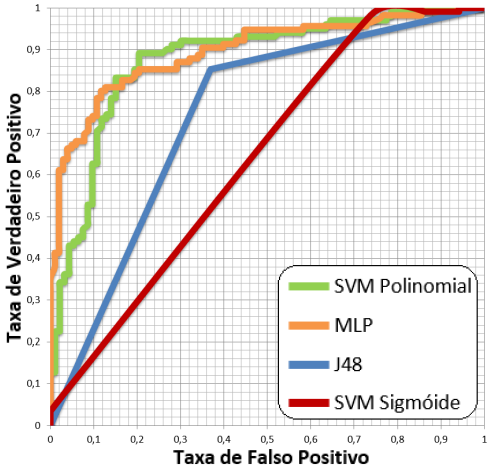

Figura 6: Curvas ROC dos classificadores utilizando apenas tons de cinza na geração do modelo de aparência.

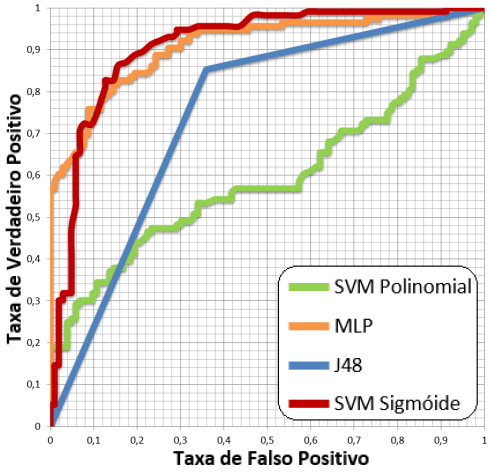

Figura 7: Curvas ROC dos classificadores utilizando características de Gabor na geração do modelo de aparência.

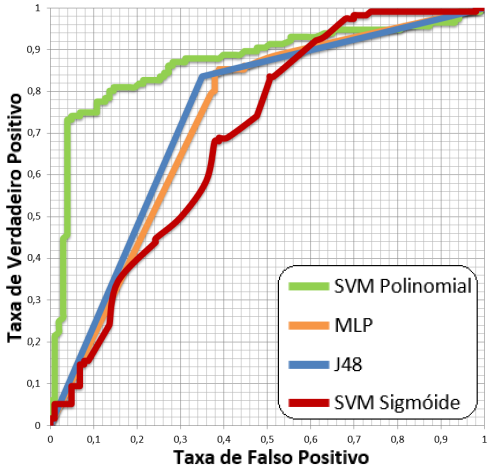

Figura 8: Curvas ROC dos classificadores utilizando características LBP na geração do modelo de aparência.

5 Conclusões

As características extraídas pelo AAM foram avaliadas para detectar a presença de capacetes de segurança de motociclistas em imagens únicas de vias urbanas. Nenhum resultado com configurações semelhantes foi encontrado na literatura para comparação. Pode-se notar que os resultados aqui obtidos com imagens individuais foram apenas ligeiramente inferiores (média precisão = 83,74%) aos resultados de trabalhos baseados em dados de *streams* de vídeo (por exemplo, em (Chiverton, 2012), precisão média = 85%), embora *streams* de vídeo possuam muito mais informações para realizar o julgamento.

Portanto, nossa abordagem revela-se promissora, uma vez que a manipulação de *streams* de vídeo é mais complexa e computacionalmente dispendiosa. Esta pesquisa prossegue agora com a investigação do uso de *color tensores* em imagens coloridas.

Agradecimentos

Agradecimentos a CAPES pelo auxílio financeiro no início da pesquisa.

Referências

Chiverton, J. (2012). Helmet presence classification with motorcycle detection and tracking, *Intelligent Transport Systems, IET* **6**(3): 259–269.

Cootes, T. F., Edwards, G. J. and Taylor, C. J. (1998). Active appearance models, *Computer Vision, 1998. European Conference on*, Springer, Freiburg, Germany, pp. 484–498.

Cootes, T. F., Taylor, C. J., Cooper, D. H. and Graham, J. (1995). Active shape models-their training and application, *Computer vision and image understanding* **61**(1): 38–59.

Edwards, G. J., Taylor, C. J. and Cootes, T. F. (1998). Interpreting face images using active appearance models, *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, IEEE, pp. 300–305.

Georghiades, A., Belhumeur, P. and Kriegman, D. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(6): 643–660.

Guibas, L. and Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of voronoi, *ACM transactions on graphics (TOG)* **4**(2): 74–123.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). The weka data mining software: an update, *ACM SIGKDD explorations newsletter* **11**(1): 10–18.

Jolliffe, I. (2002). *Principal component analysis*, Wiley Online Library, New Jersey, USA.

Ku, M.-Y., Chiu, C.-C., Chen, H.-T. and Hong, S.-H. (2008). Visual motorcycle detection and tracking algorithm, *WSEAS Trans. on Electronics* **5**(4): 121–131.

OpenCV (2014). *version 2.4.9*, Itseez, Inc., San Francisco, California.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA.

Silva, R., Aires, K., Veras, R., Santos, T., Abdala, K. and Soares, A. (2013a). Automatic detection of motorcyclists without helmet, *Computing Conference (CLEI), 2013 XXXIX Latin American*, IEEE, NaiguatÃ¡, Venezuela, pp. 1–7.

Silva, R., Aires, K., Veras, R., Santos, T., Abdala, K. and Soares, A. (2013b). Automatic motorcycle detection on public roads, *CLEI ELECTRONIC JOURNAL* **16**(3).

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features, *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 1, IEEE, Kauai, HI, USA, pp. I–511.